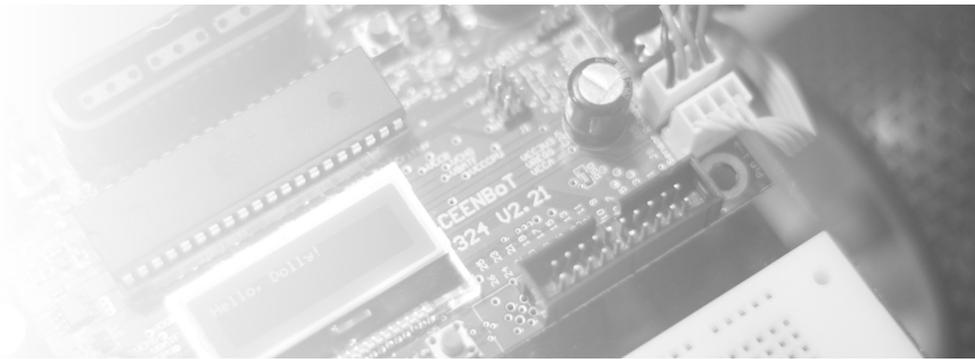


Mobile Robotics I: Lab 7b

Choose Your Own Adventure: Defining Tasks with Behavior-Based Control

**CEENBoT™ Mobile Robotics Platform Laboratory Series
CEENBoT v2.21 – '324 Platform**



Alisa N Gilmore, P.E., *Instructor, Course & Lab Developer*

The Peter Kiewit Institute of Information Science & Technology
Department of Computer & Electronics Engineering
University of Nebraska-Lincoln (Omaha Campus)

Rev 1.00

Mobile Robotics I – Choose Your Own Adventure

Purpose

The purpose of this lab is to create 3 simple tasks for your CEENBoT that are each implemented using a collection of 2 or more behaviors. First, define each task. Then, to implement each task, use existing behaviors created in earlier labs and/or define new behaviors as needed. The new behaviors are to be created using the CEENBoT's existing sensors.

Lab Objectives

By following the directions in this lab, you are expected to achieve the following:

- Define three tasks for your CEENBoT using creativity and pulling together previous functionality (behaviors) of your CEENBoT and creating new behaviors using its existing sensors.
- Implement a task by ordering several behaviors and testing and modifying them so that the behaviors individually work together to achieve an **overall emergent task behavior** in the BBC program structure.
- Discuss and document your process of creating each task, and provide a BBC diagram such as those shown in the Jones text to illustrate the behaviors used for each task and the arbitration order.
- Demonstrate the performance of each task.

Requirements

Preliminary Readings

- Read about Programming, Decomposition, Implementation, and Emergent Behavior in course texts: Mataric, Chapter 18 and Jones, Chapters 5, 6, and 8.
- Read about a practical implementation of PID controllers in: “A PID Controller for LEGO Mindstorm Robots” located here:
http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html

Required Equipment

- CEENBoT, platform '324 v2.21.
- Means to program your CEENBoT. (i.e., USB or Serial ISP programmer).

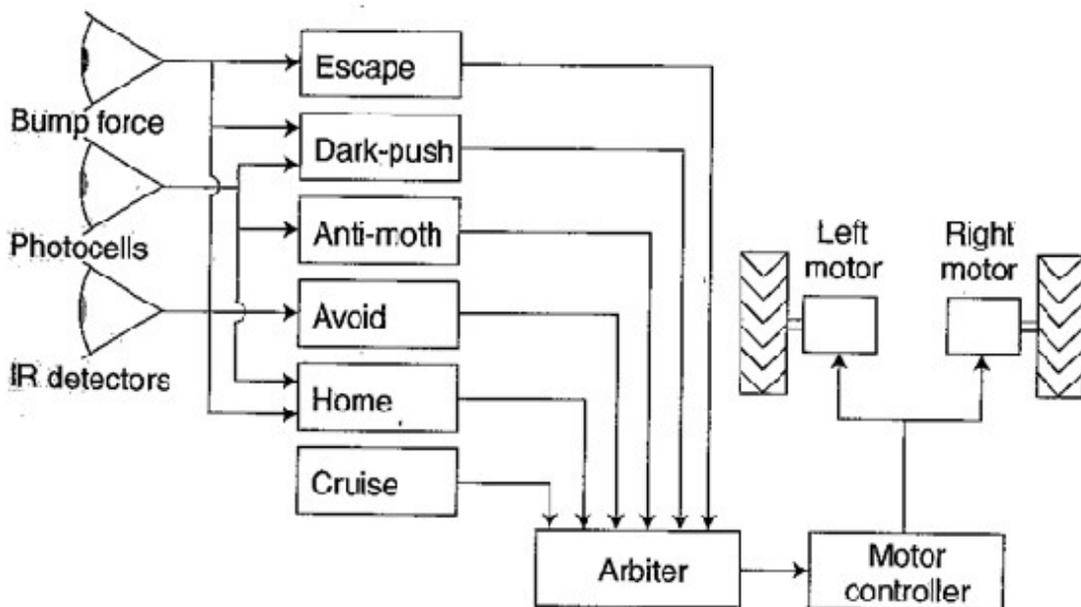
Background

This lab represents a culmination of BBC programming techniques to implement specific robot tasks.

A task achieves a desired purpose. It is implemented using a collection of behaviors in a BBC program. For example, the SodaBot had the task of moving all abandoned soda cans in an office to the recycle bin. This task was implemented using a collection of eight behaviors. Other tasks are simpler and require fewer behaviors to implement. For example, a coverage task that provides full coverage to an area in which the robot roams can be implemented using only three behaviors: Cruise, Avoid-Random, and Wall Follow.

You are to choose and define **three simple tasks** for your CEENBoT to perform. The goal of the task creation is to choose behaviors that will individually work together in the robot's environment to achieve an **overall emergent task behavior**.

Each task can be represented using a Behavior-Based Control (BBC) Diagram. It shows the sensors, behaviors and arbitration order involved. The example BBC Diagram below shows six behaviors in order of **decreasing** priority.



For each task you specify, you are to include its BBC diagram like that above.

ROBOT PROGRAMMING

You are to use your BBC C-program structure used in previous labs. For simplicity, you can use one BBC program for each task. The program will have only the behaviors appropriate for the task. Or you may include all tasks in one program, selected by push button switch input (this will require minor modifications of the program structure along the lines of making only select behaviors active that are appropriate for the selected task).

Directions

1. Define three separate simple tasks for your CEENBoT. List the purpose or desired emergent behavior for each task. Use creativity and pull together previous functionality (behaviors) of your CEENBoT and create new behaviors using existing sensors as needed to implement each task.
2. Implement each task by ordering several behaviors. Test and modify the Task's BBC program as needed so that the behaviors individually work together to achieve an **overall emergent task behavior**.
3. Once a task is working successfully, document the task created as outlined in the section below.

Questions to answer in the RESULTS section of your Report

Provide the following for each Task:

1. A description of the task, including the purpose or desired emergent behavior for the task.
2. A description of what each behavior does that makes up the task.
3. A BBC diagram that shows the relationship between the sensors and behaviors that make up each task.
4. An written overview of how the individual behaviors were designed to work together to produce the emergent task behavior, in theory (how you planned it) and in reality (what you actually observed). Did you have to make any changes to the behaviors or to how the behaviors interacted with the environment in order to achieve the task you originally desired?
5. Any additional details on how you implemented each task, where the idea for the task came from, and hurdles you had to overcome.

Deliverables

Demonstrations

You will demonstrate your robot's execution of these new capabilities during the designated class meeting or lab time.

C-code

Include a printout of your CEENBoT program. You may include *snippets* of your code and embed them in your lab report as you discuss how your program works, but a separate attachment of your entire source code must be included as part of your report.

Lab Report

The lab report should include all of the following:

- **Title Page** – Include *Course Number, Course Title, Instructor Name, Your Name, Lab Name, & Due Date*.
- **Overview** Section (a brief paragraph of what the lab was about, and the purpose behind it). Please also make sure you touch upon the following as well:
 1. Resources used: (human, text, online or otherwise).
 2. Time invested in this project.
 3. A high-level description of your robot and program.
 4. If this was a *team assignment*, state how tasks were delegated and split up among you.
 5. Known problems with your solution (e.g., the robot will not work on the *carpet*, or... it breaks if you make it go further than 5 ft (for whatever reason), etc.)
- **Background** Section (discuss some of the theory addressed in this particular lab).
- **Procedure** – Briefly describe the lab procedures for each lab – what were you asked to do? What did the process consist of? How *did you* approach it?
- **Discuss your Source Code** – Discuss and explain any relevant details behind the motivation and manner in which you have written your source code. You can *embed* [small] portions on of your code that are relevant to the discussion for clarity, but please ensure to keep a complete copy of your source code as a separate attachment (see *Source Code* section below). You may also have the reader *refer* to particular pages of your source code attachment instead.
- **Results** – Use this section to answer *questions* posted throughout your lab exercise. In particular those given out in the '*Directions*' section. In the '*Procedure*' section, feel free to refer the reader to *this* section for additional info – that is, there is no need to repeat information.
- **Conclusion** – Reiterate the objective of the lab, discuss what you have learned and any further comments you might have.
- **Source Code** – Attach your C-program also.

Mobile Robotics I – Choose Your Own Adventure

Grading

Laboratory Grading Rubric		Labs will be graded out of 30 points, unless otherwise	
Points	Lab Report	Code	Demonstration
10	Follows lab format, all sections are complete, thorough, concise. Answers all questions posed.	Properly commented, easy to follow with modular components.	Excellent work, the robot performs exactly as required.
7.5	Does not answer some of the questions or has spelling, grammatical, content errors.	Partial comments and/or not modular.	Performs most of the functionality with minor failures.
5	Multiple grammatical, format, content, spelling errors, and/or questions not answered.	No comments, not modular, not easy to follow.	Performs some of the functionality but with major failures or parts missing.
0	Not submitted or submitted late.	Not submitted or submitted late.	Meets none of the design specifications or not submitted.